COLOUR GRADIENT USING GEOMETRIC ALGEBRA

Patrice Denis, Philippe Carre

Signal Images Communication Laboratory, University of Poitiers
Bat SP2MI, Teleport 2, Bd Marie et Pierre Curie, BP 30179, 86962 Futuroscope Chasseneuil Cedex
phone: +(33)549496571, fax: +(33)549496570
email: denis@sic.sp2mi.univ-poitiers.fr, carre@sic.sp2mi.univ-poitiers.fr

ABSTRACT

This paper uses the opportunity given by the formalism of geometric algebras to perform colour image processing by describing one colour with a geometric vector in the 3D space. The paper proposes to manipulate colours in the RGB space where properties such as saturation or luminance can be described with the help of geometric operations. These geometric operations are applied on vectors encoded in $\mathscr{G}_3$. A spatial filtering operation on images with this model leads to a colour gradient that can be used for example for colour edge detection.

1. GEOMETRIC ALGEBRA

1.1 Introduction

The $\mathscr{G}_n$ geometric algebra is the 2^n-dimensional vector space extended with the geometric product from the $\mathscr{V}^n$ which is a n-dimensional vector space of real numbers. The geometric algebra manipulates entities called multivectors that are the extension of vectors to higher dimension. They are graded-algebras, that is to say scalars are represented in geometric algebra by zero-dimensional quantities (0-graded) for example. Vectors are one-dimensional directed quantities (also represented by arrows), so 1-graded. In geometric algebra, there are grade-2 entities termed bivectors which are plane segments (for example circles) endowed with orientation. In general a k-dimensional entity is known as a k-vector. For an overview on geometric algebras see [3, 6, 7] for example.

In this paper we limit the focus on $\mathscr{G}_3$ algebra because we use colour as vectors in the 3D space. In this algebra the basis $(e_0, e_1, e_2, e_3, e_{23}, e_{31}, e_{12}, e_{123})$ includes a scalar, vectors, bivectors and a trivector (3-graded) also called pseudo-scalar. The vectors are the same as those in the basis (e_1, e_2, e_3) of $\mathscr{V}^3$. The bivectors e_{23}, e_{31} and e_{12} are the geometric product of two vectors, e_2e_3, e_3e_1 and e_1e_2 respectively. The remaining parts are the scalar $e_0 = 1$ and the trivector $e_{123} = e_1e_2e_3$. Any multivector $\mathbf{A} = A_0e_0 + A_1e_1 + A_2e_2 + A_3e_3 + A_{23}e_{23} + A_{31}e_{31} + A_{12}e_{12} + A_{123}e_{123} \in \mathscr{G}_3$ is a linear combination of the basis entities also called blades. The k-grade part of a multivector $\mathbf{A}$ is given from the grade operator $\langle A\rangle_k$. For instance here, the 2-grade part of $\mathbf{A}$ is $\langle A\rangle_2 = A_{23}e_{23} + A_{31}e_{31} + A_{12}e_{12}$.

Colour image processing can be done using different colour spaces. Most devices as screens for example use the RGB space to display colours. In order to avoid numeric approximations inescapable from conversions between different colour spaces, we choose to work directly in RGB. This colour space will be considered with an Euclidian metric to allow geometric calculations. CIE L*a*b* colour space respects the human characteristics to distinct colours with Euclidian metric so to perform Euclidian calculations from RGB is an approximation. We propose to encode a $N \times M$ colour image f by its three colour components on the three vector parts of a multivector in $\mathscr{G}_3$: $\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f_1(x,y)e_1 + f_2(x,y)e_2 + f_3(x,y)e_3$. $f_1(x,y)$, $f_2(x,y)$ and $f_3(x,y)$ are the red, green and blue components of the pixel at coordinates (x,y).

1.2 Products

Products in geometric algebras are the elementary operations:

- Outer Product: This product is also called exterior product, denoted by $\wedge$. It generates the subspaces defined by linearly independent combinations of blades. For example, multiplying two linearly independent vectors gives a bivector. If this bivector is again multiplied by a linearly independent vector, the result is the pseudo-scalar.
- Inner Product: This product also called interior product or left contraction, denoted by $\rfloor$, is used to give the notion of orthogonality between two multivectors [2]. Let $\mathbf{A}$ be a a-vector and $\mathbf{B}$ be a b-vector, then $\mathbf{A}\rfloor\mathbf{B}$ is a $(b-a)$-vector subspace of $\mathbf{B}$ and orthogonal to subspace $\mathbf{A}$. If $b < a$ then $\mathbf{A}\rfloor\mathbf{B} = 0$. Two vectors are also orthogonal if their inner product is null.
- Geometric Product: This product is an associative law and distributive over the addition of multivectors. In general, the result of the geometric product is a multivector. If used on 1-vectors a and b, this is the sum of the inner and outer product: $ab = a\rfloor b + a \wedge b$. Note also that the geometric product is not commutative.
- Scalar Product: This product also called dot product, denoted by $\cdot$, is used to define distances and modulus.

1.3 Definitions

Let $\mathbf{A}$ be a multivector of $\mathscr{G}_3$.

- The reversion of $\mathbf{A}$, denoted $\widetilde{\mathbf{A}}$ is defined by $\widetilde{\mathbf{A}} = \langle A\rangle_0 + \langle A\rangle_1 - \langle A\rangle_2 - \langle A\rangle_3$. Reversion is needed to define Norm and Inverse.
- Norm or Modulus of a multivector $\mathbf{A}$ is defined by the scalar product of $\mathbf{A}$ by its reversion : $|\mathbf{A}| = \mathbf{A}\cdot\widetilde{\mathbf{A}} = \sqrt{\langle A\widetilde{A}\rangle_0}$

Let now $\mathbf{B}$ in $\mathscr{G}_3$ be the geometric product of n 1-vectors $(n \in \mathbb{N})$, $\mathbf{B}$ is called a versor.

- Every versor has an inverse which means for every $\mathbf{B}$, there is a $\mathbf{B^{-1}}$ asserting $\mathbf{BB^{-1}} = 1$; the inverse $\mathbf{B^{-1}}$ of $\mathbf{B}$ equals to $\frac{\widetilde{\mathbf{B}}}{\mathbf{B}\widetilde{\mathbf{B}}}$.
- Note that every multivector is not invertible.

2. GEOMETRIC TRANSFORMATIONS

Since the colour information of an image is encoded in the vectorial part of a $\mathcal{G}_3$ multivector, we can manipulate a colour in the 3D space with the geometric transformations given by the algebra formalism (*cf. Figure 1*):

- Translation: Let $v_1, v_2, v_t) \in \mathcal{G}_3^3$, the v_1 translation by v_2 is given by their sum $v_t = v_1 + v_2$.
- Projection: Let $(v_1, v_2, v_\parallel) \in \mathcal{G}_3^3$, the v_1 projection over v_2 is given by : $v_\parallel = (v_1 \rfloor v_2) v_2^{-1}$.
- Rejection: Let $(v_1, v_2, v_\perp) \in \mathcal{G}_3^3$, the rejection of v_1 with respect to v_2 is given by : $v_\perp = (v_1 \wedge v_2) v_2^{-1}$.
- Reflection: Let $(v_1, v_2, v_r) \in \mathcal{G}_3^3$, the reflection v_r of v_1 with respect to v_2 is given by : $v_r = v_2 v_1 v_2^{-1}$.

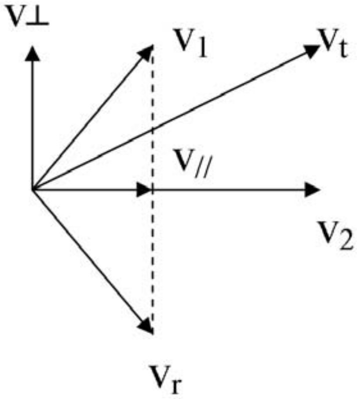


Figure 1: **Basis geometric transformations from two $\mathcal{G}_3$ vectors v_1 and v_2 :** v_t is the v_1 translation by v_2 ; $v_\parallel$ is the projection of v_1 on v_2 ; $v_\perp$ is the rejection of v_1 with respect to v_2 and v_r is the reflection of v_1 with respect to v_2.

3. COLOUR GRADIENT

From the study of the geometric transformations seen before with the vectors of $\mathcal{G}_3$, we propose to manipulate the colours of images as vector in the 3D colour space. Felsberg used geometric algebras to characterize the luminosity variations in greyscale images by a 2D analysis [5]. A first application of geometric algebras with colour images was given by Schlemmer [11]. In his paper, Schlemmer decomposed images in the LUV colour space and merged two gradients. The first one is given by a Canny method on the luminosity part of the LUV colour space. The second gradient is calculated with the two chromatic parts U and V with a 2D gradient produced from the vector field theory using the geometric algebra $\mathcal{G}_2$.

We propose in this paper to use the formalism of $\mathcal{G}_3$, that is to say to use the 3D information colour vectors, to perform a colour edge detection. In fact, in the homogenous regions of a natural colour image, pixels represented by vectors are close to their neighbours in the colour space. A colour edge can then be detected by a discontinuity in this neighbourhood. Geometric manipulations on the colour vectors of images can help us to detect such discontinuities.

A first approach was given by Sangwine in [9]. In his paper, Sangwine proposed to use the formalism of quaternions to encode the colour information of an image. This formalism is also able to manipulate vectors in the 3D space but has limits. With the definition of the convolution on a quaternionic colour image in [8, 10], he applied two specific conjugate quaternionic filters on an image. This fulfilled an average vector of every pixel in the filters neighbourhood reflected with respect to the greyscale axis. This reflection operation gives a comparison of the colour vectors in the neighbourhood of the filters (dimension of filters is 3×3). The filtered image (*cf. Figure 2*) is a greyscale image almost everywhere, because in homogeneous regions the vector sum of one pixel to its neighbours reflected by the grey axis has a low saturation. Saturation is the distance between a colour vector and the greyscale axis. This is the case of $q_4 + \mu q_3 \overline{\mu}$ for instance in Figure 3 where the quaternion μ represents the greyscale axis, $\overline{\mu}$ its conjugate and $\mu q_3 \overline{\mu}$ the reflection of q_3 with respect to μ [9]. However, if you compare pixels like q_1 and q_2 for example in Figure 3, you should detect a colour edge as they are in colour opposition. Therefore, they present a vector sum far from the grey axis. Edges are thus coloured due to this high distance. The limit of this method is that the filters can be applied horizontally from left to right and from right to left for example but without giving the same results. In Figure 3, the vector sum $q_2 + \mu q_1 \overline{\mu}$ is different from the vector sum $q_1 + \mu q_2 \overline{\mu}$ produced when the filters are applied from opposite side (ie. leftwise or rightwise). An edge with the Sangwine method will be detected by full coloured pixels in opposition to homogenous regions which will be represented by grey pixels.

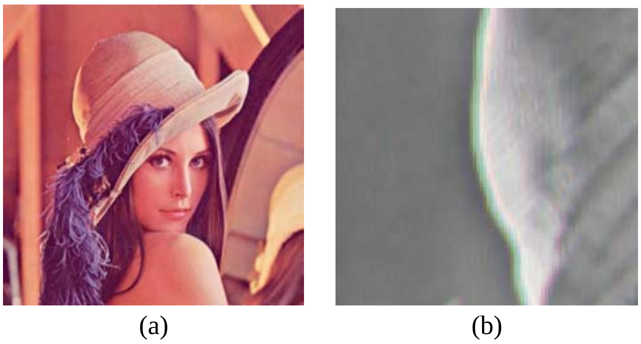


Figure 2: **Sangwine edge detector result :** (a) original image; (b) zoom on the hat edge of the (a) picture after application of Sangwine's filters from right to left horizontally.

In this paper, we propose to detect edges by a colour gradient. The method of Sangwine gives us the starting point. In fact, we have a map of every pixel reflected by the grey axis and compared to its neighbours. The more the vector result is far from the grey axis, the more it is far as well from its neighbours. Convolution product is defined in [4] with geometric algebras. The filtered image is given by $f'(x,y) = \sum\sum l(\tau_1,\tau_2) f((x-\tau_1),(y-\tau_2)) r(\tau_1,\tau_2)$ with Sangwine's generalized filters to $\mathcal{G}_3$:

$$l = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ \mu & \mu & \mu \end{bmatrix} r = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ \mu^{-1} & \mu^{-1} & \mu^{-1} \end{bmatrix}$$

We suggest to determine the distance of Sangwine comparison vector sum to the grey axis: its saturation. This

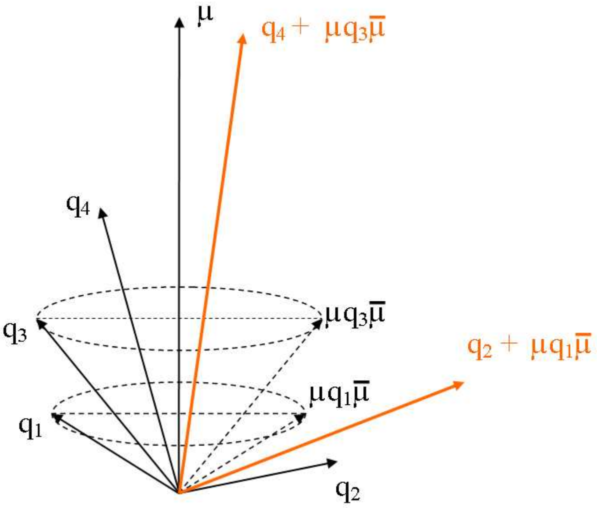


Figure 3: **Sangwine's edge detector scheme :** μ is the grey axis; $\mu q_1\overline{\mu}$ (resp. $\mu q_3\overline{\mu}$) is the reflected vector of q_1 (resp. q_3) with respect to μ; the comparison vector between q_1 and q_2 (resp. between q_3 and q_4) is given by $q_2 + \mu q_1\overline{\mu}$ (resp. $q_4 + \mu q_3\overline{\mu}$); $q_4 + \mu q_3\overline{\mu}$ is near the grey axis so the colour seems grey but $q_2 + \mu q_1\overline{\mu}$ is far from the grey axis so Sangwine's filter has detected an edge as this vector is more coloured.

distance S is the norm of the rejection of the colour vector sum v with respect to the grey axis μ. We then get it by the formula of the rejection: $S = |(v \wedge \mu)\mu^{-1}|$ (*cf. Figure 4*). The same colours v_1 and v_2 can give two different colour vectors $v_{sum_1} = v_2 + \mu v_1\overline{\mu}$ and $v_{sum_2} = vq_1 + \mu v_2\overline{\mu}$ by the Sangwine's method. Note that our approach, which gives a saturation gradient, is independent from the path (leftwise or rightwise) applied to convolute the filters as the distance $S_1 = |(v_{sum_1} \wedge \mu)\mu^{-1}| = |(v_{sum_2} \wedge \mu)\mu^{-1}|$. This saturation filtering is applied to the horizontal, vertical and both diagonal directions by $\mathscr{G}_3$ convolution product. The maximum of these directional gradients is then selected to make the final colour gradient filter by maximum saturation. This final filtering operation is not linear only because the "maximum" operator interferes.

As the method described before is a saturation gradient, a problem appears: discontinuities in colours are not just saturation ones. Indeed, two different colours can have the same saturation; fortunately they can be disjoined by their difference of luminance.

To improve our algorithm we use the opportunity given by the embedding of the Geometric Algebra. In fact, applying the geometric product: $f''(x,y) = f(x,y)\mu$ on every pixel of the original image gives us a geometric comparison of the pixels to the greyscale axis. Then the result of this operation adds luminance information. It is the sum of two parts: $f(x,y) \wedge \mu + f(x,y)\rfloor\mu$. The first part, $f(x,y) \wedge \mu$, is bivectorial and allows to compare the geometry between the colour vector and μ. When this part is null for example, the colour vector is collinear to the greyscale axis. The second part,$f(x,y)\rfloor\mu$, is a scalar, and the projection of the colour vector over the grey axis : its intensity.

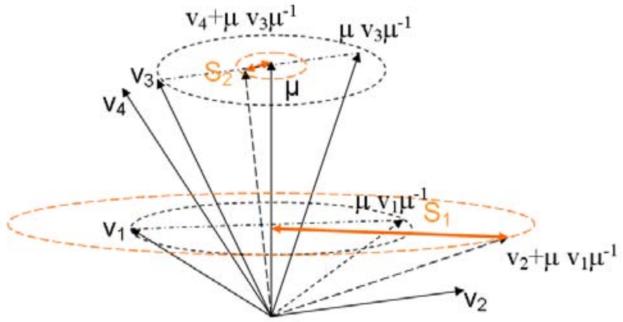
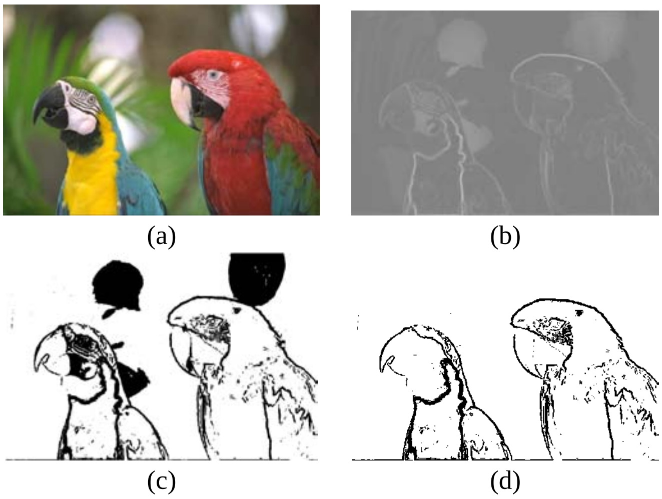


Figure 4: **Saturation gradient :** $\mu v_1\overline{\mu}$ (resp. $\mu v_3\mu^{-1}$) is the reflected vector of v_1 (resp. v_3) with respect to the grey axis μ; our comparison vector between v_1 and v_2 (resp. between v_3 and v_4) gives the distance $S = |(v_{sum} \wedge \mu)\mu^{-1}|$ of Sangwine's vector sum $v_{sum_1} = v_2 + \mu v_1\mu^{-1}$ (resp. $v_{sum_2} = v_4 + \mu v_3\mu^{-1}$) from the grey axis (orange arrows); An edge is detected by a high saturation like the orange arrow between μ and $v_2 + \mu q_1\mu^{-1}$.

A boolean mask is produced by thresholding the bivectorial part where its norm is very small to keep the coordinates where the colour vector is mere collinear to μ. We then process a Sobel filtering on the scalar part to produce a luminance gradient. Next, we apply the previous mask on this gradient to keep only the needed luminance information. In fact when a colour vector is quite collinear to the greyscale axis, the saturation gradient given by the previous method is not appropriate. Finally this is merged to the saturation gradient in order to also detect greyscale variations. We see for example in Figure 5 that the method using the luminance information merged to the maximum saturation gradient is more efficient as there are more details on the left parrot head. There are as well big achromatic regions behind the parrots detected with this method that are not with the saturation gradient method only.

This geometric product used in the second part of our algorithm could have been processed with a different μ vector, it would have given the same geometric comparison to the colour vector chosen. As our previous gradient was based on saturation, we needed to add luminance information and that is why we chose to fill μ with the greyscale axis.

As the perception of colour is linked to the Human Visual System, it may be difficult to compare results in colour image processing with greyscale quantitative evaluation methods. We could have compared our results to colour images segmentation databases but the goal of this paper was more to show how using Geometric Algebra could help to perform a colour gradient. We can nevertheless compare our method visually to other ones in Figure 6. Di Zenzo method [12] is based on a vectorial approach (c). The marginal method (b) is a fusion of three different gradients done on the three colour channels of the image. It is quite effective on this picture. Results show that our method detects colour edges properly with the house image where walls, roof and sky are well separated (e). Because the luminance in (e) is considered, we see for instance the details of the window's house, the drainpipe and the bottom of the roof that do not appear on the maximum saturation approach (d). The Carron method (f)


(a) (b)

(c) (d)

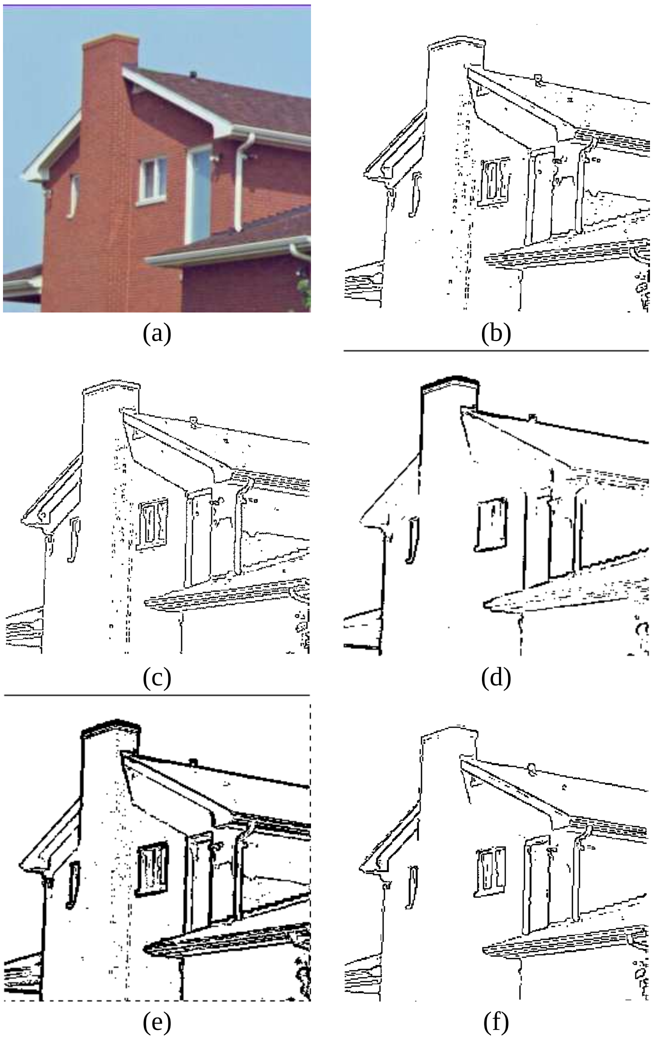
Figure 5: **Colour gradient examples :** (a) original image ; (b) colour gradient with a luminance adjustment; (c) thresholding on the colour gradient ; (d) thresholding on the maximum saturation gradient (without the modification of the filters).

is based on a hue marginal gradient but it is completed by luminance and saturation gradients when hue is not enough [1].

The edges obtained by our approach (d and e) are much thicker than in all the other methods. The Carron approach gives the best result but it is based on hue, luminance and saturation variations whereas our method uses only the two latter. Our method gives about the same results than the Di Zenzo one with thicker edges. As we work directly in the RGB colour space, we do not need to determine a geometric distance between colours; we use a saturation distance plus a luminance one when necessary.

To conclude, we propose an original method based on geometric algebra to detect colour edges. This formalism allows to manipulate and to compare the colours as vectors in the 3D space. This method gives a geometrical gradient that can detect colour edges properly compared to already known approaches like Di Zenzo or Carron ones. It uses RGB colour space but detects saturation and luminance variations with geometrical operations allowed by $\mathcal{G}_3$. As geometric algebras are not limited to three dimensions, an extension to N-dimension from this work is possible. Multiband images as those produced by satellite sensors for example are being studied to be processed by geometrical filters in $\mathcal{G}_n$.

The authors thank Poitou-Charentes Region board which has funded this work.


(a) (b)

(c) (d)

(e) (f)

Figure 6: **Comparison between different edge detectors :** (a) original image ; (b) marginal method result; (c) result from the Di Zenzo method ; (d) result from the maximum saturation gradient ; (e) result from the proposed method ; (f) result from the Carron method ;

REFERENCES

[1] T. Carron, ”Segmentation d’images couleur dans la base Teinte-Luminance-Saturation : approche numérique et symbolique“, Thèse de doctorat, Université de Savoie, december 1995.

[2] L. Dorst, ”The Inner Products of Geometric Algebra “, in *Applications of Geometric Algebra in Computer Science and Engineering* : Dorst, Doran and Lasenby eds., Birkhauser, pp. 35–45, 2002.

[3] L. Dorst and St. Mann, ”Geometric algebra:a computational framework for geometrical applications (part I:algebra) “, in *Computer Graphics and Applications, IEEE*, pp. 24–31, May/June 2002.

[4] J. Ebling and G. Scheuermann, ”Clifford Fourier transform on vector fields “, in *IEEE Transactions on Visualization and Computer Graphics*, vol. 11,4, pp. 469–479, July/August 2005.

[5] M. Felsberg, ”Low Level Image Processing with the Structure Multivector “, PhD Thesis, March 2002.

[6] D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus* : Kluwer Academic Publishers, 1984.

[7] D. Hestenes, *New Foundations for Classical Mechanics: Fundamental Theories of Physics* : Kluwer Academic Publishers, 2nd edition, 1999.

[8] C.E. Moxey, S.T. Sangwine and T.A. Ell, ”Hypercomplex Correlation Techniques for Vector Images “in *IEEE Transactions on Signal Processing*, vol. 51, pp.

1941–1953, July 2003.

[9] S. J. Sangwine, ”Colour image edge detector based on quaternion convolution “, in *Electronics Letters*, vol. 34,(10), pp. 969–971, May 14 1998.

[10] S. J. Sangwine, ”Colour in Image Processing “, in *Electronics and Communication Engineering Journal*, vol. 12,5, pp. 211–219, Oct. 2000.

[11] M. Schlemmer, H. Hagens, I. Hotz and B. Hamann, ”Clifford Pattern Matching for Color Image Edge Detection “, in *Visualization of Large and Unstructured Data Sets, GI-Edition Lecture Notes in Informatics (LNI)*, vol. S-4, pp. 47–58, Oct. 2006.

[12] S. Di Zenzo, ”A note on the gradient of multi-image “, in *Computer Vision, Graphics, and Image Processing*, vol. 33, pp. 116–125,1986.